

IDENTIFYING CUSTOMER INTEREST FROM SURVEILLANCE CAMERA BASED ON DEEP LEARNING

Anjali T P

MES20MCA-2008

Product Owner: Mr Balachandran KP

TABLE OF CONTENTS

- 1.Introduction
- 2.Modules
- 3.Data Flow Diagram
- 4.Table Design
- 5.Developing Environment
- 6.Product Backlog
- 7.User stories
- 8.Project Plan
- 9.Sprint plans
- 10.Sprint Actual

INTRODUCTION

Identifying customer's interests is valuable as it intuitively represents the product the customer wants. It can also be an effective marketing strategy for determining potential customers. Therefore, large retail vendors like Walmart and Costco analyze customer purchase history to identify customer interest. However, purchase history alone cannot fully determine how much interest in the product a customer has other than what they have purchased. In other words, products that the customer does not purchase but are interested can never be identified. This project focuses on identifying a customer's interest based on behaviors from surveillance cameras. We detect the customer's gaze direction as this behavior accurately reflects customer interest in a particular product

MODULES

Module 1: Admin:

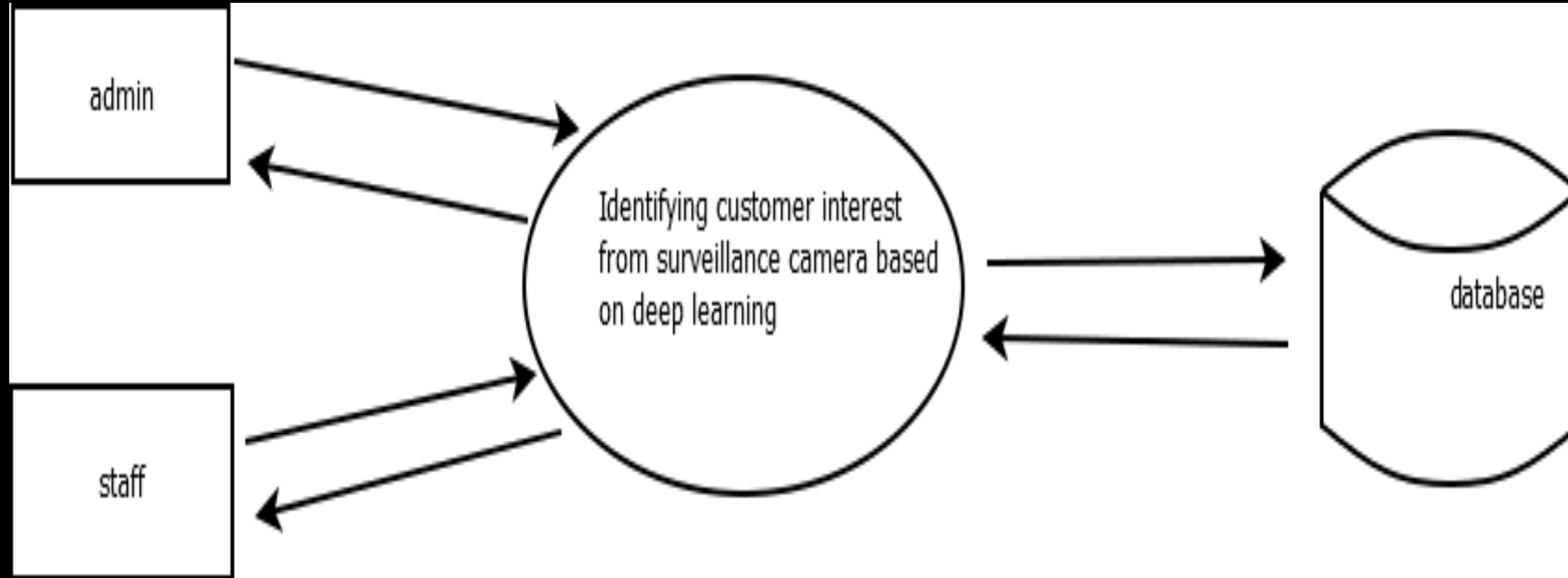
- Add and manage Staffs
- Add and manage camera
- Send Notification to Staff
- View Notification
- Assign work to staff
- View work status

Module 2: Staff:

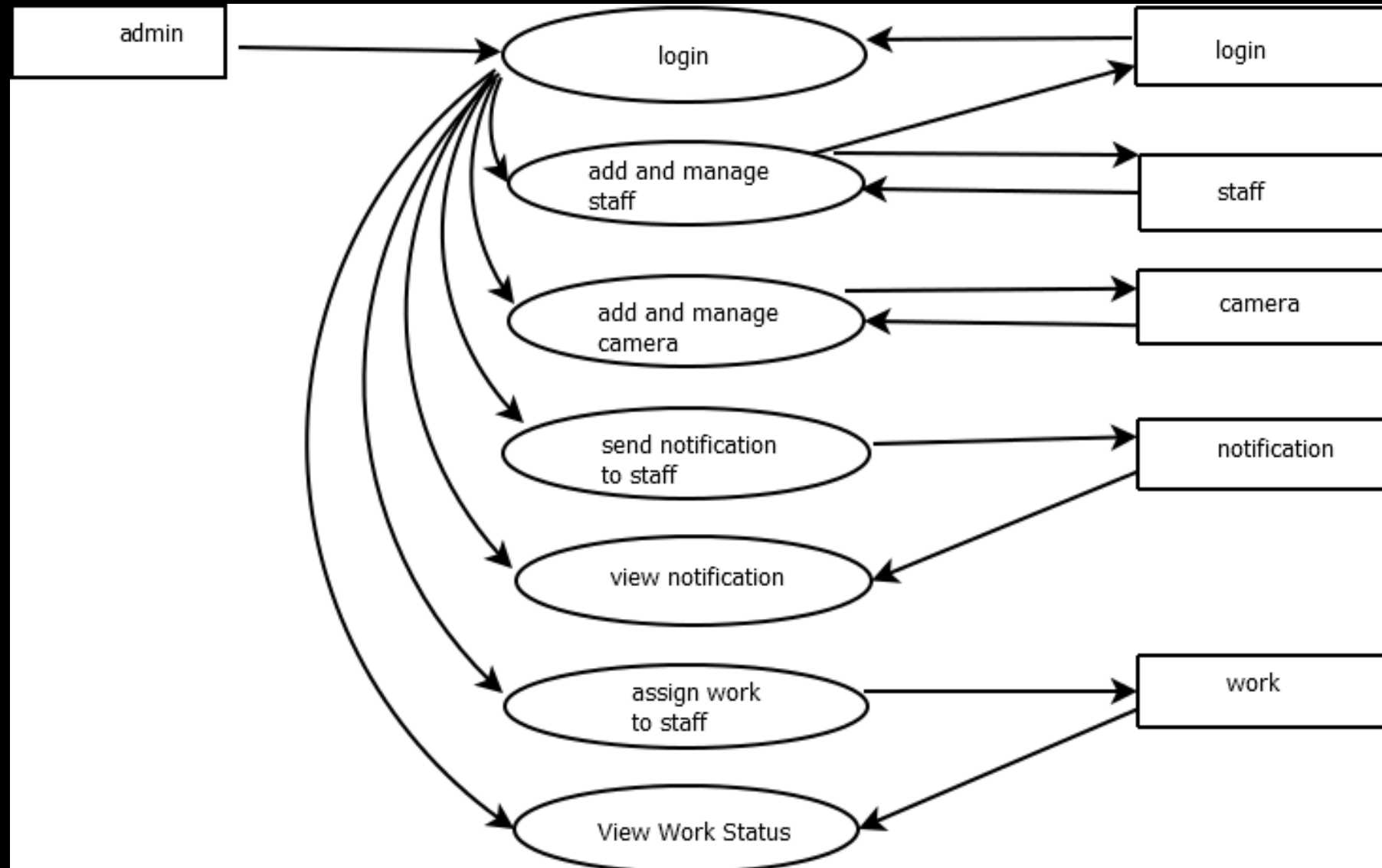
- Login
- View works and update status
- View notification from admin
- View notification from camera

DATA FLOW DIAGRAM

LEVEL 0



LEVEL 1.1



LEVEL 1.2

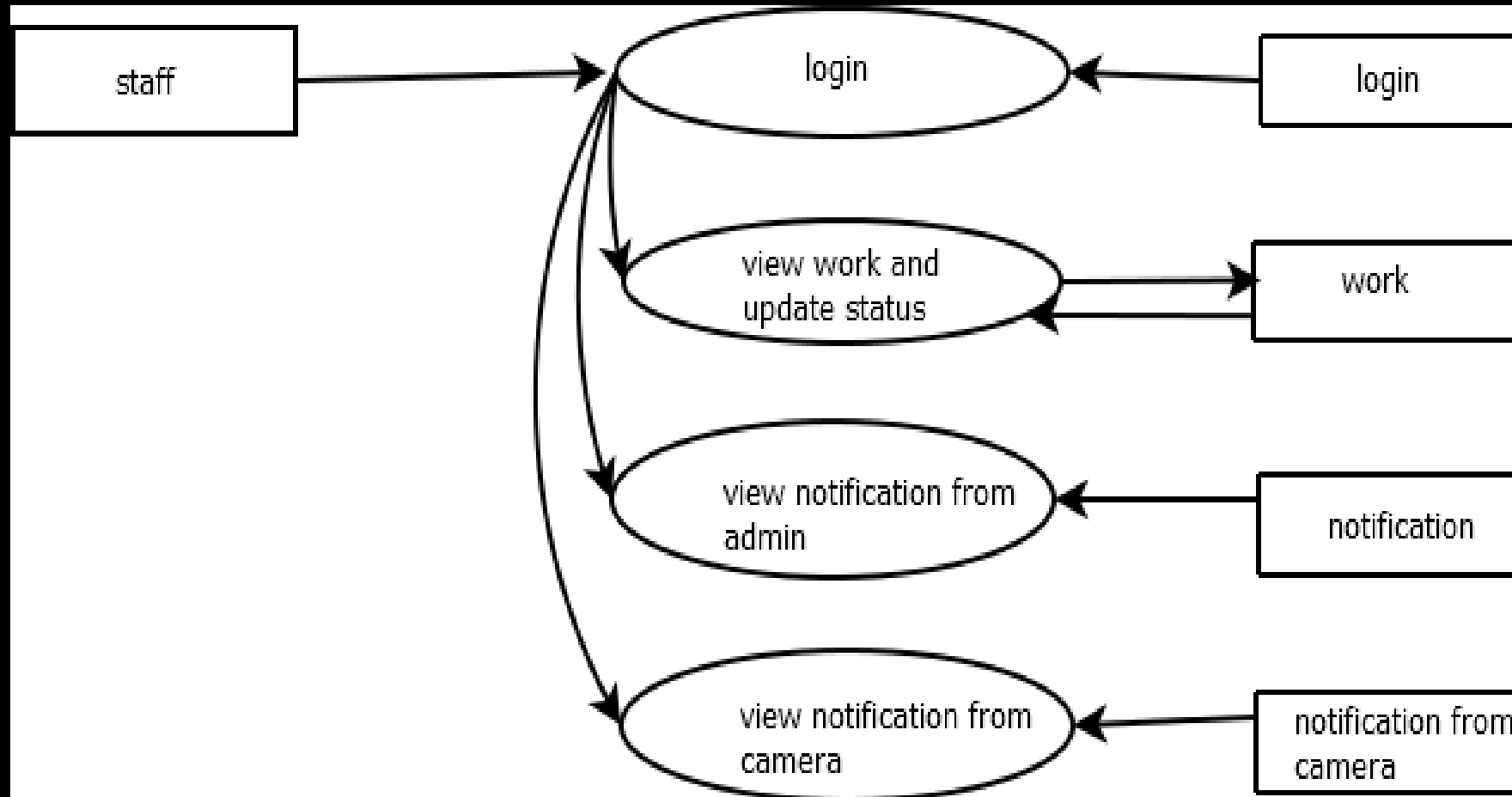


TABLE DESIGN

Login

[illegible]

Staff

[illegible]

Camera

| Columns | | | | | | | | | | | |
|---|-----------|--------|---------|-------------------------------------|-------------------------------------|--------------------------|-------------------------------------|--------------------------|--------------------------|---------|--|
| 2 Indexes | | | | | | | | | | | |
| 3 Foreign Keys | | | | | | | | | | | |
| 4 Advanced | | | | | | | | | | | |
| 5 SQL Preview | | | | | | | | | | | |
| + - ▲ ▼ | | | | | | | | | | | |
| <input checked="" type="checkbox"/> Hide language options | | | | | | | | | | | |
| <input type="checkbox"/> Column Name | Data Type | Length | Default | PK? | Not Null? | Unsigned? | Auto Incr? | Zerofill? | On Update | Comment | |
| <input type="checkbox"/> id | int | 11 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | | |
| <input type="checkbox"/> cameranumber | varchar | 20 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | | |
| <input type="checkbox"/> | | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | | |

Notification

| Columns | | | | | | | | | | | |
|---|-----------|--------|---------|-------------------------------------|-------------------------------------|--------------------------|-------------------------------------|--------------------------|--------------------------|---------|--|
| 2 Indexes | | | | | | | | | | | |
| 3 Foreign Keys | | | | | | | | | | | |
| 4 Advanced | | | | | | | | | | | |
| 5 SQL Preview | | | | | | | | | | | |
| + - ▲ ▼ | | | | | | | | | | | |
| <input checked="" type="checkbox"/> Hide language options | | | | | | | | | | | |
| <input type="checkbox"/> Column Name | Data Type | Length | Default | PK? | Not Null? | Unsigned? | Auto Incr? | Zerofill? | On Update | Comment | |
| <input type="checkbox"/> id | int | 11 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | | |
| <input type="checkbox"/> notification | varchar | 40 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | | |
| <input type="checkbox"/> date | date | | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | | |
| <input type="checkbox"/> camid | int | 11 | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | | |
| <input type="checkbox"/> | | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | | |

Work

[illegible]

DEVELOPING ENVIRONMENTS

- Languages used: Python
- Front End : HTML, CSS, JAVASCRIPT
- Backend : MySQL
- Data set : Facial emotion recognition (FER) data set(Kaggle dataset)
- OS : Windows 7 or Above, Android
- Platform used : JetBrains, PyCharm, Android Studio
- Frame work : Flask
- Technology :Python, Java
- Algorithm : Haar Cascade Algorithm(This algorithm has 4 steps)

1. Haar Feature Selection

Objects are classified on very simple features as a feature to encode ad-hoc domain knowledge and operate much faster than pixel system. The feature is similar to haar filters, hence the name 'Haar'. An example of these features is a 2-rectangle feature, defined as the difference of the sum of pixels of area inside the rectangle, which can be any position and scale within the original image. 3-rectangle and 4-rectangle features are also used here.

2. Integral Image Representation

The Value of any point in an Integral Image, is the sum of all the pixels above and left of that point. An Integral Image can be calculated efficiently in one pass over the image.

3. Adaboost Training

For a window of 24x24 pixels, there can be about 162,336 possible features that would be very expensive to evaluate. Hence AdaBoost algorithm is used to train the classifier with only the best features.

4. Cascade Classifier Architecture

A cascade classifier refers to the concatenation of several classifiers arranged in successive order. It makes large numbers of small decisions as to whether it's the object or not. The structure of the cascade classifier is of a degenerate decision tree.

USER STORIES

| UserStoryID | As a type of user | I want to <Perform Some Task> | So that I can <Achieve Some Goal> |
|-------------|-------------------|----------------------------------|---|
| 1 | Admin | login | login successful with correct username and password |
| 2 | Admin | Add and manage staff | Add ,view,edit,delete the staffs |
| 3 | Admin | Add and manage camera | Add ,edit ,delete the camera number |
| 4 | Admin | Send notification to staff | Send notification to the staff |
| 5 | Admin | View notification | View the notification from camera |
| 6 | Admin | Assign work to staff | Assigned work to individual staff |
| 7 | Admin | View work status | View the work status |
| 8 | staff | login | login successful with correct username and password |
| 9 | Staff | View work and update | View the work details and update |
| 10 | Staff | View notification from admin | View the notification from admin |
| 11 | Staff | View notification from camera | Camera notification is viewed |

PRODUCT BACKLOG

| User Story ID | Priority <High/Medium/Low> | Size (Hours) | Sprint <#> | Status <Planned/In progress/Completed> | Release Date | Release Goal |
|---------------|-------------------------------|-----------------|---------------|---|--------------|----------------------------------|
| 1 | Medium | 2 | 1 | Completed | 08/01/2022 | Table design |
| 2 | High | 3 | | Completed | 08/01/2022 | Form design |
| 3 | High | 5 | | Completed | 08/01/2022 | Basic coding |
| 4 | High | 5 | 2 | Planned | | Data set creation |
| 5 | Medium | 5 | | Planned | | Detection of face |
| 6 | High | 5 | 3 | Planned | | customer's gaze direction method |
| 7 | Medium | 5 | | Planned | | identify customer interest |
| 8 | Medium | 5 | 4 | Planned | | Testing data |
| 9 | High | 5 | | Planned | | Output generation |

PROJECT PLAN

| User Story ID | Task Name | Start Date | End Date | Days | Status |
|---------------|-----------|------------|------------|------|-----------|
| 1 | Sprint 1 | 26/12/2021 | 28/12/2021 | 2 | completed |
| 2 | | 29/12/2021 | 31/12/2021 | 3 | completed |
| 3 | | 03/12/2021 | 08/01/2022 | 5 | completed |
| 4 | Sprint 2 | 09/01/2022 | 16/01/2022 | 8 | Planned |
| 5 | | 18/01/2022 | 22/01/2022 | 5 | Planned |
| 6 | Sprint 3 | 23/01/2022 | 27/01/2022 | 5 | Planned |
| 7 | | 30/01/2022 | 05/02/2022 | 7 | Planned |
| 8 | Sprint 4 | 06/02/2022 | 10/02/2022 | 5 | Planned |
| 9 | | 16/02/2022 | 19/02/2022 | 4 | Planned |

SPRINT BACKLOG PLAN

| Backlog item | Status and completion date | Original estimate in hours | Day1 | Day2 | Day3 | Day4 | Day5 | Day6 | Day7 | Day8 | Day9 | Day10 | Day11 | Day12 | Day13 | Day14 |
|----------------------------------|----------------------------|----------------------------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|
| User story#1,#2,#3 | | | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs |
| Table design | 28/12/2021 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Form design | 31/12/2021 | 3 | 0 | | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Basic coding | 08/01/2022 | 5 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| User story #4,#5 | | | | | | | | | | | | | | | | |
| Data set creation | 16/01/2022 | 5 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Detection of face | 22/01/2022 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| User story #6,#7 | | | | | | | | | | | | | | | | |
| Customer's gaze direction method | 27/01/2022 | 5 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Identify customer interest | 05/02/2022 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| User story #8,#9 | | | | | | | | | | | | | | | | |
| Testing data | 10/02/2022 | 5 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Output generation | 19/02/2022 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 |
| Total | | 40 | 4 | 4 | 3 | 3 | 4 | 3 | 3 | 5 | 4 | 2 | 2 | 2 | 1 | 0 |

SPRINT ACTUAL

| Backlog item | Status and completion date | Original estimate in hours | Day1 | Day2 | Day3 | Day4 | Day5 | Day6 | Day7 | Day8 | Day9 | Day10 | Day11 | Day12 | Day13 | Day14 |
|----------------------------------|----------------------------|----------------------------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|
| User story#1,#2,#3 | | | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs | hrs |
| Table design | 28/12/2021 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Form design | 31/12/2021 | 3 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Basic coding | 08/01/2022 | 5 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| User story #4,#5 | | | | | | | | | | | | | | | | |
| Data set creation | | | | | | | | | | | | | | | | |
| Detection of face | | | | | | | | | | | | | | | | |
| User story #6,#7 | | | | | | | | | | | | | | | | |
| Customer's gaze direction method | | | | | | | | | | | | | | | | |
| Identify customer interest | | | | | | | | | | | | | | | | |
| User story #8,#9 | | | | | | | | | | | | | | | | |
| Testing data | | | | | | | | | | | | | | | | |
| Output generation | | | | | | | | | | | | | | | | |
| Total | | 10 | 1 | 1 | 2 | 1 | 0 | 1 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 |

THANK YOU